

HERE: A Web Based Environment for Requirements Engineering

Papaioannou V. and Theodoulidis B.

Department of Computation, UMIST

PO Box 88, Manchester M60 1QD, United Kingdom, tel. +44

161 200 3309, fax +44 161 200 3324

Email: {vp,babis}@sna.co.umist.ac.uk

Abstract

This paper introduces the main ideas regarding the generic *HERE* (*Hypermedia Environment for Requirements Engineering*) framework and computer environment. The main goal of this approach is to target the intrinsic complexity that is associated with the *Requirements Engineering* products. This is achieved by providing support for the effective analysis, management and co-ordination of the information gathered throughout the early stages of a system development into a single, consistent, manageable entity. The *HERE* mechanisms provided cater for this management both at the conceptual level and at the tool support and functionality level. Our theoretical framework is mainly based on the innovative use of *Ontologies* for supporting the *Requirements Engineering* process and products. Finally, in terms of tool support we achieve the implementation of our framework with the use of an advanced environment based on a *Hypermedia-World Wide Web* setting.

Keywords

Requirements Engineering, Information Systems, Ontologies, Hypermedia

1. INTRODUCTION

Whilst the importance of *Information Systems* technology has been realised since the late 70's - beginning of the 80's, their growth is more apparent than ever before today and it is expected to rise even more within the next years. Parallel to the expansion of these systems was the increase of their complexity, which also resulted to a great deal of complexity of their respective development methods [Loucopoulos & Karakostas, 1995], [Pohl, 1994]. Moreover, the importance of Requirements Engineering (RE) was recognised, since various studies have shown that errors in the early stages of information systems development, i.e. establishment of requirements and analysis, were the hardest to fix later and resulted to a numerous of unsuccessful or even undelivered systems [Ramamoorthy et al, 1984], [Boehm, 1989], [SEI, 1995], [ARIANES, 1996], [IWSSD-8, 1996], [LAS, 1993].

Despite the great interest in the area of requirements engineering, the success was mostly on the first rather than the second term. In other words RE practitioners still find it very hard to engineer requirements. Additionally, despite the demonstration of interesting approaches, still there is no universal standard or description of the RE process and the full scope is not yet established. The HERE project was originally motivated from the need to provide a hypermedia environment to enhance the RE process. The research undertaken ended up with a hypermedia environment and its underlying HERE framework. According to our work, the crux of the requirements engineering problem is complexity. Complexity may come into different forms:

- Product complexity is associated with RE products that may exist in various forms, media, etc. and may contain information covering different levels of knowledge, different views, at varying levels of formality. RE Information is very dynamic and may become difficult to inspect, manage, trace and maintain.
- Organisational Complexity has to deal with the various groups of stakeholders that are involved in the RE process each one with different views, interests, domain knowledge, objectives (often conflicting), etc. This diversity is also reflected in the RE products as mentioned above.
- Process Complexity that has to do with the fact that there is no one single way of coping with the issue of requirements, nor there is one single standard to acquire, analyse, model, verify, and measure requirements.

The HERE framework targets at the complexity that hardens the analysis, communication and management of the RE knowledge, which is essentially encapsulated within the RE products, not by adding an extra methodology but a supporting framework.

The main mechanism provided by HERE is the ability to keep information about the Models, Knowledge, and Data concerning the problem domain under one manageable set of explicit shareable conceptualisations named *Ontologies*. Moreover, *links* are specified among different components of this knowledge. This knowledge is stored within a *Hypermedia Database* repository. Then a Hypermedia Engine, consisting of nodes and links, interfaces this information to its end user.

Section two presents our motivation for our approach, section three analyses how the HERE framework supports the RE process, section four overviews the role of ontologies in the HERE project and section five introduces us to our view of link models. Finally, sections six and seven present our environment implementation.

2. MOTIVATIONS FOR THE HERE APPROACH

It is expected that Requirements Engineering processes may be initiated because of a problem or a need in the Universe of Discourse (UoD). Regarding Information Systems, reasons might fall in one of the following situations: an emerging need for a new information system, a need for change in the current system, and finally a need to record the current status of the domain of interest.

Independently of any specific methods that might be used we may expect a wide range of information sources or RE products: Informal information (from interviews, meetings, meetings, personal communication), existing repositories, documentation, designs, requirements specifications, models, etc.

The information and artefacts described above are expected to be communicated to a number of stakeholders: *requirement engineers, future system end users, domain experts, analysts, developers, testers, and maintainers, etc.*

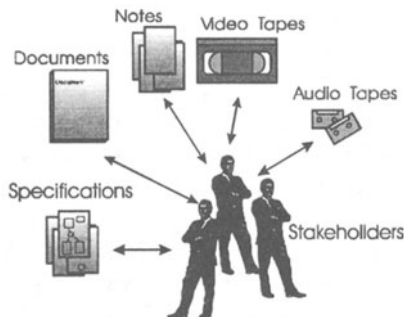


Figure 1: Information sources and its recipients in the HERE scenario.

Klaus Pohl ([Pohl, 1994]) identifies three main goals for the Requirements Engineering (RE) process:

- improving an opaque system comprehension into a complete system specification;
- transforming informal knowledge into formal representations;
- gaining a common agreement on the specification out of the personal views;

Though the RE concerns and goals appear to be quite simple, the process for addressing them tends to be very complex. More specifically, we may regard the following of issues - problems that arise during the RE activities [Loucopoulos & Karakostas, 1995], [Macaulay, 1996], [Pohl, 1994]:

- *Varying levels of information formality.* Information may be hard to analyse because formal specifications have to be produced and cross-referenced with informal sources. Important information involving the UoD, e.g. rationale, is missed out when they are transformed into models and vice versa. Models and formal specification may become obsolete if changes take place in the UoD.
- *Varying levels of information abstraction.* When information to be analysed spans to more than one abstraction level e.g. Enterprise Strategy, Enterprise Processes, Enterprise Information System, then the analysis may also become complex and dependencies between one level and another may be lost. Changes at a higher level may fail to be reflected to lower levels and vice versa.
- *Multiplicity of representation formalisms.* Information conflicts, inconsistencies and overload may occur because of the large number of existing methodologies or formalisms that may be applied. For example parts of the system development may have been developed using structured methodologies and specifications whilst others may follow the object oriented paradigm.
- *Multiple views.* Information is partitioned into different viewpoints depending on the concern of the stakeholders involved. Each view is associated with its own context and vocabulary. For example manager may be interested in strategic goal models whilst database operators may be interested in transaction models of their database. Furthermore, differing views may introduce gaps in terminology and semantic conflicts.
- *Multiple formats and mediums.* The parties involved in the RE process are overwhelmed with many of the elements of information sources as described above. As mentioned earlier this information may come from various sources, various viewpoints, and in many different mediums. Consequently, stakeholders and especially analysts may easily get confused because of the amount and disparity of information. From a practical point of view, requirements stakeholders have to cope with the management and

maintenance of a large number of documents stored in many different mediums.

- *Fragmentation of information.* Analysts have to combine all available information and their dependencies but information elements have to be combined from many information sources. Thus the analysis process is hindered by the information fragmentation whilst important information elements may be missed. Traceability may become a very hard task whilst engineering legacy is lost from project to projects or even within the same project.

Based on this scenario we worked towards the establishment of RE support environment and conceptual framework, which primarily addresses the issues for analysis, management and communication as described above as well as other RE issues as presented earlier. The next sections will present the HERE framework and its implementation.

3. THE HERE SUPPORT FOR RE

According to the scenario of the previous section, we realise that the information of a development project, and especially information which is aimed at the stakeholders involved in a RE process, is multifaceted, voluminous, and comes in many different forms in order to cover all aspects concerning the problem domain. The HERE approach is mainly concerned with the data, products, information, and knowledge that are acquired throughout the RE activities as described above.

In HERE, we view this information essentially covering three main aspects of the Universe of Discourse:

- Knowledge about the *Domain requirements* and needs: enterprise requirements, user requirements, non-functional requirements, etc.
- Knowledge about the *Enterprise Domain*: the enterprise structure, its processes, actors, goals, etc.
- Knowledge about the *Information System*: system requirements, system design, database design, process design, etc.

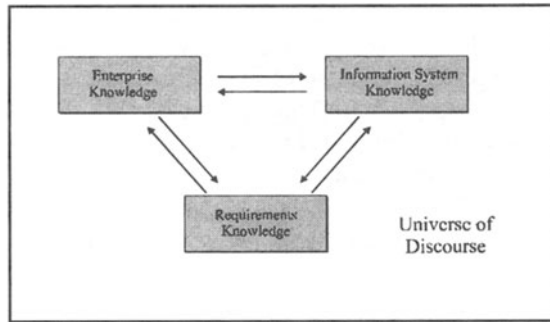


Figure 2: Areas of UoD knowledge.

The main problem is how to organise the output and feedback originating from the RE activities into a concrete manageable set of specifications which will empower the RE stakeholders with the ability to analyse, manage, and communicate these specifications and knowledge in order to achieve their objectives. This problem is addressed by the HERE approach both at the conceptual and the implementation level.

At the conceptual level, in the HERE approach we aim at two primary objectives:

- The integration and coupling of the disparate information elements which are produced during the RE processes into an explicit, unified, clarified, and defragmented representation. In HERE the above knowledge is explicitly expressed into sets of models and meta-models which describe concepts and meta-concepts respectively regarding the Universe of Discourse, in a modular, structured manner.
- The explicit specification of meta-models and models which describe the interconnections among these knowledge elements.

In HERE, these two goals are accomplished with the use of *Ontologies*:

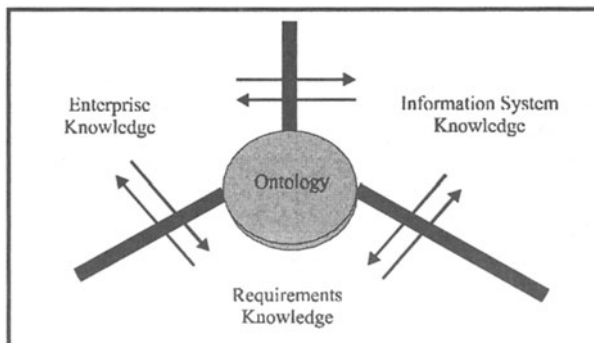


Figure 3: Integrating the problem domain knowledge in HERE.

At the implementation level, we aim at the management and communication of the above knowledge elements by providing tool support that is implemented as a hypermedia system built on top of an object-oriented multimedia database. The hypermedia system is based on the HTML standard and the database contents can be communicated over the World Wide Web. Analysis of the implementation issues is presented in sections six and seven.

The next figure presents the overall architecture of the HERE environment:

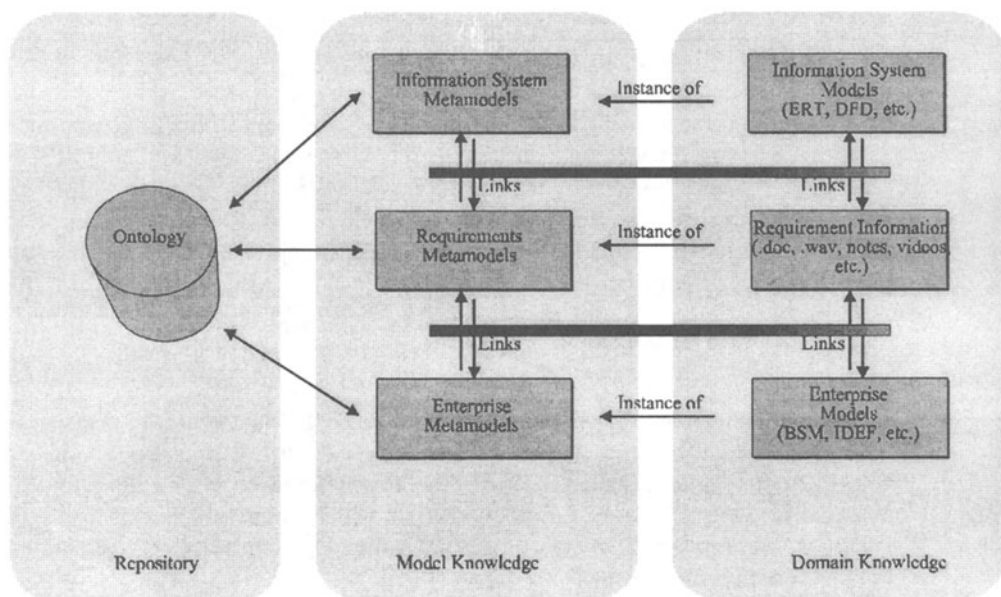


Figure 4: The HERE architecture.

In the HERE architecture, we may identify three main components:

- The *Model Knowledge* component: Includes the knowledge that describes models that may be used in order to specify domains of interest. In other words, this model contains meta-model information. This information spans over the three levels of UoD knowledge as described earlier, i.e. Enterprise, Requirements, and System knowledge.
- The *Domain Knowledge* component: This component includes the knowledge that is modelled for a specific domain. The models are instances of model specifications in the Model Knowledge component. According to their

content and purpose, they also span over the Enterprise, Requirements, and System knowledge.

- The *Repository* component: It holds all the above specifications in the form of Ontologies along with all relevant data that contribute to the overall knowledge of the UoD.

4. THE ROLE OF ONTOLOGIES IN HERE

One of the main priorities of the HERE approach was to address the problem of heterogeneous representations of all the information elements produced as a result of the RE processes. At the implementation level, the utilisation of a multimedia capable database management system was the answer. However, heterogeneity also existed at the conceptual level. Models and other specification mechanisms can be regarded the primary forms of representation abstractions of the problem domain. Since HERE aims at the effective analysis, management, and communication of all these elements, a solid paradigm had to be adopted to form the basis of our approach. Recently the concept of Ontology emerged as a research topic mainly in the areas of *Knowledge Base System* integration and *Artificial Intelligence Knowledge Representation* approaches. Although the term *Ontology* is also appeared as a conference theme in the *Information - Software System* development and other relative communities, only few of their approaches have actually used Ontologies solely for representing or integrating Enterprise knowledge.

But what exactly is an ontology? So far, researchers admit that the term is quite difficult to grasp and neither agree completely nor they can express in absolute terms the ontology concept.

An *Ontology* can be defined as:

An explicit specification of conceptualisation. A conceptualisation is based on a body of formally represented knowledge; a set of definitions that allows one to construct expressions about some application domain (i.e. an abstract, simplified view of the world that we wish to represent for some purpose) [Gruber, 1993], [Schreiber et al, 1995], [Wielinga et al, 1994].

The term has its origin in philosophy, where an ontology is a systematic account of existence. It was then embraced in AI & Knowledge Based systems in order to support the sharing and reuse of formally represented knowledge among different systems in a commonly defined vocabulary. Ontologies have been developed for various domains.

Examples of ontologies can mainly be found in the Library provided by the Stanford University's Knowledge Systems Lab (KSL). Specific examples are the bibliographic-data ontology which defines the terms used for describing bibliographic references, the theory "BASIC-MATRIX-ALGEBRA" which attempts to capture basic concepts in linear algebra, with emphasis on matrix operations, the theory "JOB-ASSIGNMENT-TASK" which defines a task ontology for the job assignment task, the theory "THERMAL-SYSTEM" which is a domain theory for engineering thermodynamics, and many others [Farquhar et al, 1995]. Other efforts include the Enterprise Ontology which is currently being developed at the AIAI (Artificial Intelligence Applications Institute, University of Edinburgh) [Fraser, 1994].

As far as representation languages is concerned many approaches used a different way of coding ontologies. Representations include: Prolog, Conceptual Graphs, L-Lilog, Loom, Classic, CLIPS, CORBA's IDL, KIF, CML, EXPRESS, Ontolingua. Representations can also vary from informal (natural language) to semi formal and rigorously formal (formal semantics, theorems and proofs of soundness and completeness) [Farquhar et al, 1995], [Ushold, 1995].

In our approach the concept of Ontologies was adopted within the context of *Requirements Engineering* and the main reasons are summarised below.

In HERE Ontology models differ slightly from traditional conceptual models. This is because Ontologies are driven from the need to describe the elements that exist in the UoD in a generic and explicit manner, so future references have to commit to these conceptualisations. Traditional conceptual models on the other hand are more specific in abstracting and describing the problem domain in the context of developing a system. In this sense, the content of Ontologies may be viewed as a superset of these models; for example, a specific data model of a problem domain may be based on the Ontology of the domain of interest. Thus, we view Ontologies as the ideal result of the RE process where there is an effective exploitation of the domain concepts and knowledge elements.

Another special characteristic of Ontologies is that they are explicit specifications of conceptualisations that are commonly agreed. Thus in HERE we view Ontologies within the RE context and regard them to include the agreement dimension:

$$\textit{Ontology} = \textit{Model} + \textit{Agreement}$$

In this view Ontologies serve as more enhanced abstractions of reality which demonstrate a higher level of conformance to the three dimensions of

requirements engineering, namely *specification*, *representation*, and *agreement*, as described by K. Pohl ([Pohl, 1994]) and may form a reliable basis of reference, communication, knowledge sharing and reuse.

Ontologies are inventories of the kinds of things that are presumed to exist in a given domain together with a formal description of the salient properties of those things and the salient relations that hold among them. In HERE Ontologies are used to hold the knowledge about models and model instances of the UoD in one unified representation. Thus by using a common representation ontologies become a computational medium to integrate all the domain information to describe domain knowledge in two primary realms: The *Metamodel* realm and the *Instance* realm:

a) The *Metamodel* realm;

- It is realised with the *model* component in HERE and it holds meta-level information, i.e. meta-models of the models that are used to describe the domain of interest. Examples are the ERT¹ model ontology, the Enterprise Ontology², etc. In this way, Ontologies form one single repository of domain knowledge that can serve as a reference point which can be reasoned throughout the development lifecycle.
- Apart from the model knowledge, in this realm we also describe the *Link* models, which hold the information about the links that are pertinent to these models at the meta-level. Consequently, at the instantiation level, in the Domain component, the models automatically realise these meta-level links. Examples of Link models are the ERT to Enterprise intermodel links, etc.

b) the *Instance* realm;

- The instance realm is realised in the *domain* component of HERE where the meta-level ontology descriptions are actually instantiated with knowledge elements concerning the domain of interest. Examples are The ERT schema of the UMIST Admissions or the Enterprise Ontology of the UMIST Admissions.
- The Model Link Ontologies are also instantiated at this level, providing the actual links that exist among the instance level information elements. E.g. The links that relate the ERT schema of the UMIST Admissions to the Enterprise Ontology of the UMIST Admissions

1 ERT stands for Entity Relationship and Time Model [Theodoulidis et al, 1991].

2 The Enterprise Ontology is investigated by the Enterprise project [Fraser, 1994].

Another important point is that the Ontological descriptions are expressed in a *modular, expandable, reusable* manner:

- *Modular*: One might concentrate his interest and reason about specific parts of an ontology concerning specific contexts. Specific parts of the ontology may be imported or exported.
- *Expandable*: Ontologies are not static. They are refined over time and may be expanded. Ontology elements may be added or deleted on demand.
- *Reusable*: Ontologies are regarded to be quite generic and commonly agreed, therefore they can be reused. It is in fact one of the purposes of ontologies to be the primary mediums for knowledge sharing. Once an ontology has been developed for some domain then future developments have to commit to this ontology.

Once the input is transformed into ontologies then the review, reasoning and management of the RE information becomes feasible as ontologies can then be browsed examined and managed. The fact that ontologies describe the model specifications used does not imply that the HERE ontologies replace these formalisms. The idea is for the HERE framework to provide support for the existing methodologies and formalisms and not the introduction of a new one.

Ontologies in HERE are expressed according to the Object Oriented paradigm. In this way the process of conceptualising and developing an ontology is more familiar within the requirements engineering and / or systems - software engineering communities and these representations are more related to the future system development. Other benefits are the intrinsic advantages of object oriented approaches, such as easy mapping from the problem domain to the object-oriented specification, modularity, structuring, encapsulation, inheritance, etc. As it is understood that object oriented concepts are very common towards implementation, this must not be confused with our motivations. In our framework object concepts are mainly viewed as representation rather than implementation constructs.

In HERE the choice of representation relies on the object model as defined by the ODMG group [ODMG, 1997]. The ODMG is a consortium of object-oriented database management system (ODBMS) vendors and interested parties working on standards (ODMG 2.0) to allow portability of customer software across ODBMS products. The ODMG 2.0 is the integration of: Objects (OMG), Databases (SQL), and Languages (C++, Smalltalk, and Java). The standard builds upon each of these standards to define a standard for developing ODBMS applications.

5. LINK MODELS

When developing Information Systems and throughout RE phases, many different methods, models and viewpoints might be used [Pohl, 1994], [Karakostas & Loucopoulos, 1989], [Nuseibeh et al, 1994]. We advocate that the conceptualisation of the relationships between the models and model constructs used to record the problem domain knowledge, are as important as the elements of this knowledge themselves.

The information gathered in a large development project may be of any capacity. The fact that this information is modelled and recorded is only the first step. The crux of the problem is how this information will actually be used, especially during the following development phases. Therefore, it is essential to provide means for managing and communicating this information. The models themselves may provide some support for that. But relations between knowledge elements are more generic and more useful since they enable the reasoning of the modelled domains from many alternative meaningful viewpoints. In addition, relations are the only means for providing a true image of the modelled knowledge, since concepts like specific model relations e.g. possible dependencies between two models may now reveal the actual situation.

Some of the motivations for such an approach in HERE are:

- To ensure *completeness*. Information elements and artefacts ranging from original requirements specifications to system design products, can be cross referenced and therefore provide the full view of the problem domain knowledge rather than scattered pieces of information. Presentation of information in this way can also be checked for completeness to ensure that important information is not missed out.
- *Consistency*. When different views of the knowledge of the problem domain are examined at the same time, then possible inconsistencies can be discovered and eliminated.
- *Impact Analysis*. When we establish the interrelations of information elements then we are able to reason about possible impacts among them. Thus, future changes are implemented in a justifiable manner in the system and integration of new components is performed with consistency.
- *Quality*. The products and artefacts of the early phases of a system development plus the traceability links improve the quality of produced specifications (requirements specifications, design specifications, implementation specification) and thus improving the quality of the implemented system.
- *System Development Information Management*. Traceability enables stakeholders to locate and effectively manage this information. Rationale and

design / implementation decisions are preserved throughout the system development phases.

- *Customer verification and satisfaction.* Information traceability provides the means by which the customer can verify his requirements against the proposed system and thus traceability ensures customer satisfaction throughout the whole development. It ensures system acceptance since it ensures that all requirements are fulfilled or no unnecessary system elements are added to customers expense.
- *Maintenance.* This is the hardest and most expensive period of a system because it happens after its delivery and spans throughout the systems lifetime. Traceability is the only mechanism that can relate parts of the implemented system with higher level design and requirement specifications. Thus, it enhances the maintainer's potential for a systematic and effective maintenance.

On the other hand, links in this sense may be quite generic. The notion of relationships that may exist within a model (*intramodel* links) or between different models (*intermodel*) is quite abstract and always depends on the context in which these relations take place. Consequently in different contexts, we are interested in creating and looking at different relations. Moreover, links need to be defined according to the needs of the people participating in the process.

In the HERE framework, we define a *working set of Link Ontology Models*. Some of these models can be found in the current bibliography where they are defined either explicitly or implicitly. Explicit modeling and classification of the links makes their capture and retrieval more systematic in order to avoid a chaotic situation where everything is linked arbitrarily to everything producing confusion and overhead. In short, our Ontology consists of a hierarchy of three types of classified links:

- *Model Links.* They conceptualise the semantic links that exist implicitly or explicitly among the same or different model concepts.
- *Traceability Links.* They conceptualise the links that refer to the traceability among RE artefacts.
- *User Defined Links.* This is the third option in the hierarchy in order to allow the HERE user to define his own link models or incorporate there any other models that do not fit in our classification.

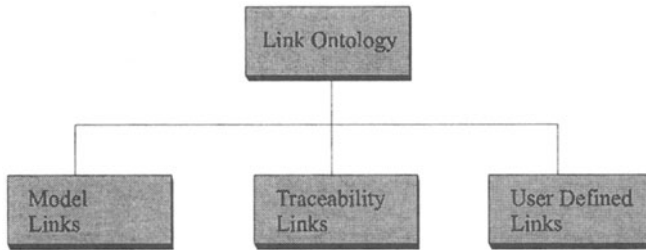


Figure 5: A hierarchy of Link Ontologies.

For each of the categories above, the HERE users may either adopt an existing link model that fit in these categories or invent their own link model. In our implemented framework, these links are stored along with the rest information elements and they are translated into hyperlinks connecting them accordingly.

6. THE HERE ENVIRONMENT SUPPORT

The discussion so far has only been referred to the theoretical aspects of our framework. Thus, we have seen how it is possible to provide both intentional and extensional specifications of the problem domain knowledge. In a real situation in order to take the full advantage of our work, our framework has to be supported by the HERE environment. The HERE environment realises the framework objectives into a tangible toolkit, which exploits its full potential.

- The top level objectives of the HERE environment are:
- The ability to define and manage Ontologies as meta-level definitions of object models, which can be instantiated later.
- The ability to interactively examine these the definitions; the ability to query ontology models.
- The ability to instantiate Ontology meta-models and populate-manage the project database.
- The ability to interactively examine and query about the model instances.

For these purposes, we have constructed a prototype environment, which demonstrates our approach. Based on our objectives of the HERE environment we may identify the following top level requirements:

- *Ontology Definition:* Ontologies have to be developed at the intentional layer, to describe models, design, or specification formalisms including domain ontologies and other information models that are used in order to maintain the knowledge about the Universe of Discourse. Therefore, the environment

should accept and store object definitions in ODL, which will form the domain ontologies. This requires the study of all the possible ODL concepts such as class-subclass definitions, their properties, their methods, etc. Ontology definitions have to be stored in the environment repository.

- *HERE repository population:* After defining the schema of concept elements describing the problem domain and their interrelations then it is possible to populate the HERE database with model instances, ontology instances, and all relative RE data which actually describe the domain of interest (extension layer). In implementation terms, the repository formed is an object-oriented database, which holds all the elements described so far. The environment should also support the management of this information, like updates, new insertions, and deletions of elements.
- *Ontology Repository Inspection:* it is desirable for the HERE users to interactively examine the Ontology definitions. The inspection involves both the meta-level and the instance realm. The repository examination involves interactive queries and browsing of its contents.

The above requirements are realised via three main tool components:

- The *Ontology Editor:* This component is responsible for defining, specifying ontologies. The users of this editor are mainly ontology engineers who develop the necessary ontologies collaborating with the domain experts who are responsible for holding the knowledge concerning the problem domain.
- The *Ontology Manager:* It supports the population of the HERE repository with data at the domain level, that is all instances of models as described in the metamodel section (model component), and all relative RE data such as documents, videos, etc. Transactions such as insertion, deletion and update of new elements are also supported. The users of this component are expected to be requirements engineers, system designers, etc.
- The *Ontology Browser:* It enables the examination and reasoning of the HERE repository. This component is hypermedia based, built on top of the object oriented database management system. It allows the easy exploration of the here repository elements through interactive browsing and database queries. This component is used by everyone who needs access to the central repository information.

Apart these top-level requirements regarding the functionality of the HERE environment another two main implementation requirements may be identified:

The need for an *object-oriented repository:*

- Since we adopted an object oriented framework for describing ontologies, the best way to implement our approach is the choice of an object oriented

database management system able to accept and store ODL schema definitions which correspond to meta-level Ontology descriptions. Database instances will then correspond to Ontology instances.

- Our need for storing, manipulating and retrieving elements other than text and numbers, e.g. audio, video also supports this argument.

This requirement is satisfied with the use of the *POET Object-Oriented Database Management System* (OO DBMS) from POET Software Corporation [POET 1]. Poet is ODMG compatible and accepts ODL definitions, among other features, and thus making it an optimum environment for developing PC based Object Oriented database applications.

The need for a collaborative *hypermedia* setting:

- In our framework, we want to be able to store and display various document types ranging from diagrams up to pictures audio and video and possibly synchronising them together. A hypermedia environment is the most efficient way for displaying the contents of our object repository.
- Central to our approach is the expression or the relationships that may be identified between the information elements stored in our repository. A hypermedia setting is the most natural way of translating these links into an implemented environment.
- The cognitive dimension of hypermedia environments allows the knowledge to be easily inspected and communicated. Hypermedia systems are well known for these attributes as they allow the inspection-transfer of large amounts of information following a series of nodes linked together with hyperlinks.

Regarding the hypermedia aspect of our implementation, our choice relied on the *WWW (World Wide Web)* technology. Below we summarise our main reasons for this choice:

- The popularity of WWW applications and the Internet as a medium for research and development in the recent years.
- The facilitation of access to hypermedia documents based on the HTML specification.
- The familiarity of end users to this category of computer–interfaces and interaction which imposes no overhead to the user. No special requirements are needed in the client side and portability is ensured to almost every platform available today. Moreover, it imposes no overhead to the developer, as there are no special requirements for developing such applications.
- The ability for the parties involved to work in a collaborative distributed environment. Because RE entails the collaboration of many parties, it would be desirable for the tool to allow any disparate stakeholders to participate in the process.

Therefore, our implementation top-level architecture is WWW based as depicted in the next figure:

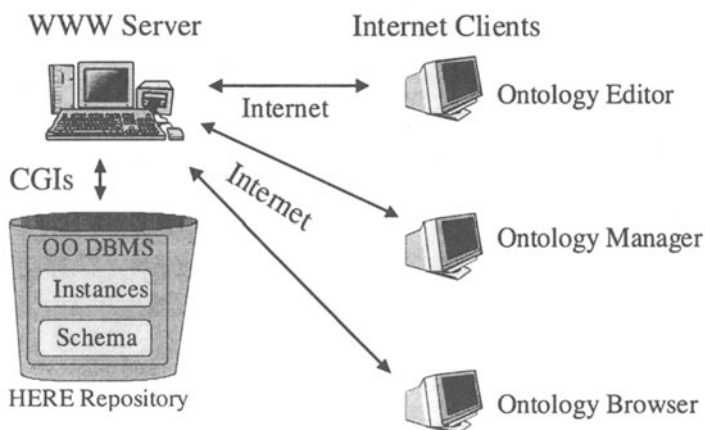


Figure 6: The HERE implementation overview.

The main idea is that we have a central object oriented database management system which serves as our repository. This repository contains both meta-level (database schema) and instance-level descriptions (database instances). The repository is connected to a WWW server. Clients can communicate with the server via the Internet. All clients use a WWW interface (web browser) which allows them to interact with the central HERE repository. The clients can perform the basic operations required, i.e. define and manipulate ontology schemas (Ontology Editor), populate ontology instances (Ontology Manager), and finally inspect the repository contents (Ontology Browser). The server can fulfil the requests of its clients by running a set of Common Gateway Interface (CGI) programs. These programs accept the requests on the server and return HTML output to the clients.

The next section emphasises on the functionality of the HERE environment.

7. THE HERE ENVIRONMENT FUNCTIONALITY

The Ontology Editor

The Ontology Editor is responsible for enabling users to specify store and manipulate ontology definitions. The definitions have to conform to the ODL

specifications. The operations are performed after requests from the client machines that communicate with the central repository server via the Internet. The main idea here is that users are working on distinct projects that are associated with a set of ontologies as displayed in the following picture:



Figure 7: Selecting a Project.

The user can choose a project to work on, from a list of existing projects or select a new project. Each project is supposed to be associated with a set of ontologies, so the user can select any of them to edit:

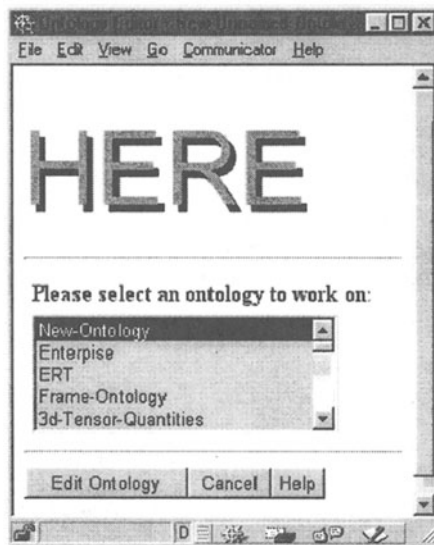


Figure 8: Selecting an Ontology to edit.

The user can include or exclude certain ontologies from a project or even construct a new ontology according to his/her needs. The next figure depicts the screen for creating a new ontology:

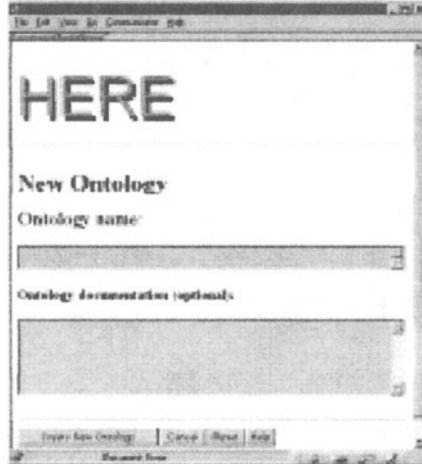


Figure 9: Creating a new Ontology.

When constructing a new ontology the main tasks include the complete definition of the ontology in ODL terms. This can be performed using the Web interface to submit the ontology specifications to the main repository. The following screen show how this can be done using our current implementation:

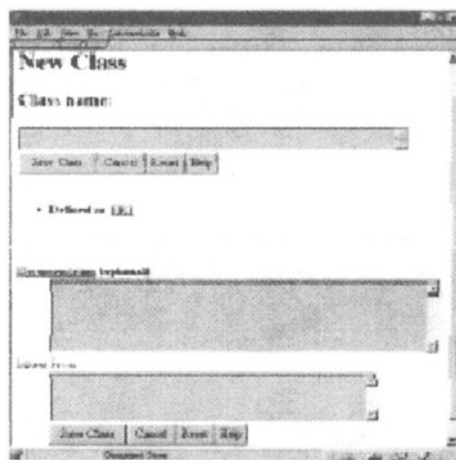
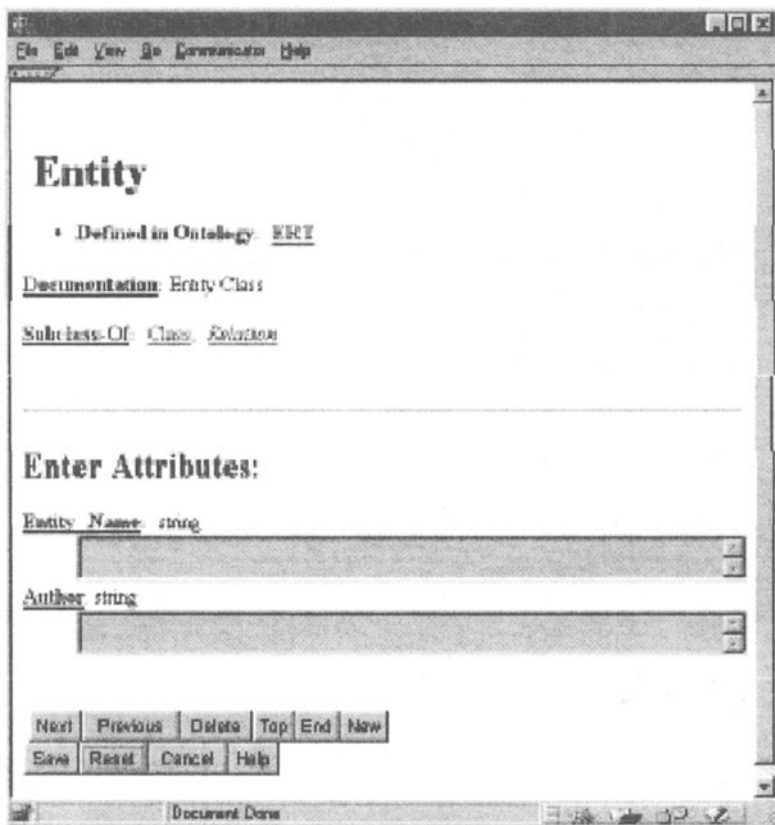


Figure 10: Creating a new Class

The Ontology Manager

After defining the Ontologies at the meta-level then, it is possible to populate the HERE repository with the instances corresponding to the domain of interest. This is achieved with the Ontology Manager Component, which adds, deletes, or updates the ontology instances. For example, instantiating the ERT model for a specific domain involves the instantiation of all objects as defined in the ERT model. Our Web based interface can also cater for this population:



The screenshot shows a web browser window with a title bar and a menu bar. The main content area displays the following information:

- Entity**
- Defined in Ontology: [ERT](#)
- Documentation: [Entity Class](#)
- Subclass-Of: [Class](#), [Relation](#)

Below this information is a section titled **Enter Attributes:** with two input fields:

- Entity Name: string
- Author: string

At the bottom of the form, there are two rows of buttons:

- Row 1: Next, Previous, Delete, Top, End, New
- Row 2: Save, Reset, Cancel, Help

The status bar at the bottom of the window shows "Document Done".

Figure 11: Inserting a new Entity instance into the repository.

It also must be noted that in our prototype implementation filetypes that cannot be populated in this manner, e.g. multimedia files can be “filled in” the repository by using the file location (Uniform Resource Location, URL).

The Ontology Browser

The Ontology browser component implements our requirement for inspecting and reasoning about ontologies and their instances describing a domain of interest. The inspection can be carried out either as interactive browsing or by using queries. In all occasions, hyperlinks can connect related elements according to their context. The inspection covers two main areas of the repository: the meta-level definitions and the instance-level definitions realised in the model and domain components of the HERE architecture respectively.

When browsing the Ontologies at the intension layer, users are presented with object hierarchies, which essentially form the overall ontology for a specific project. Then by clicking on individual objects they can follow a path of examining more and more elementary components of this ontology. An example of such browsing is presented in the next figure:

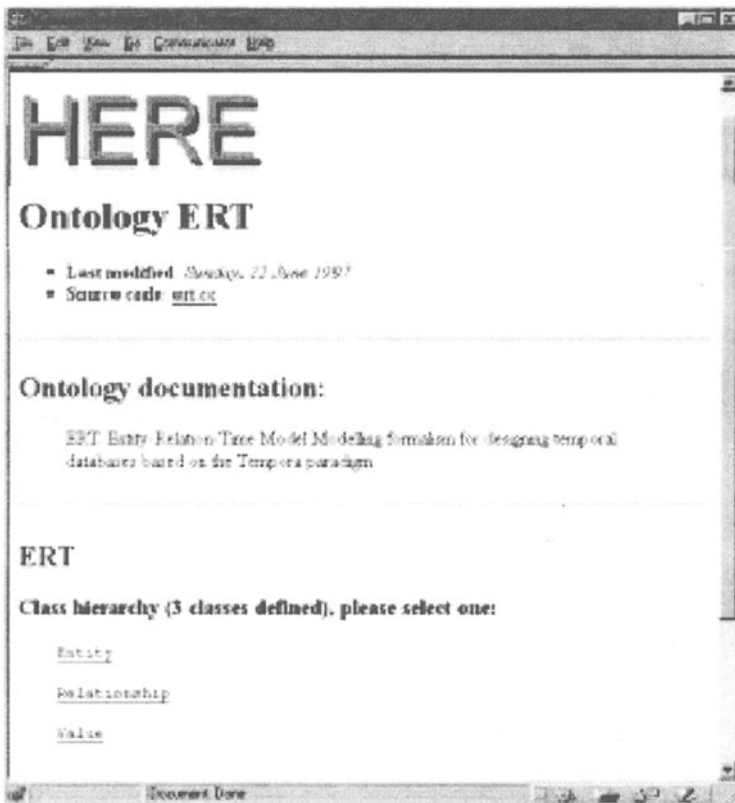


Figure 12: Browsing the ERT Ontology.

Here we can view the ERT ontology, its documentation and its main attributes, i.e. Entities, Relationships and Values. If a class link is followed then the user is presented with the class attributes, data types, etc.:

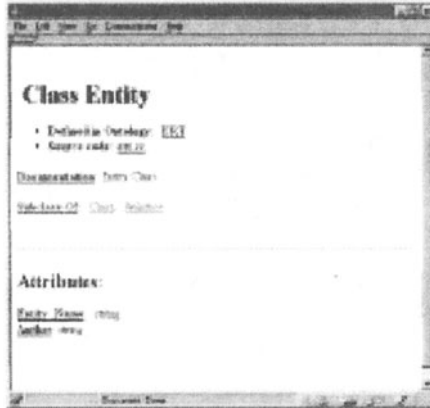


Figure 13: Browsing the Entity Class definition.

Apart from browsing at the intension-level, a user may also be involved in the browsing of ontology instances:

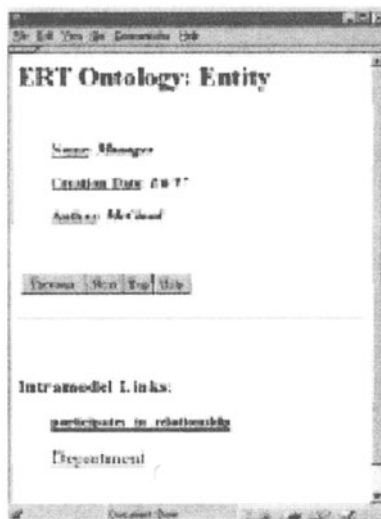


Figure 14: Browsing an instance of the Entity Class.

The screen above shows an example of browsing the instances stored in the main repository together with the possible “intra-links” that connect the objects of interest with other related objects.

Finally, queries of the repository can result to a series of objects to be examined based on the query criteria. The queries return a set of hyperlinks to other repository objects based on the query criteria:

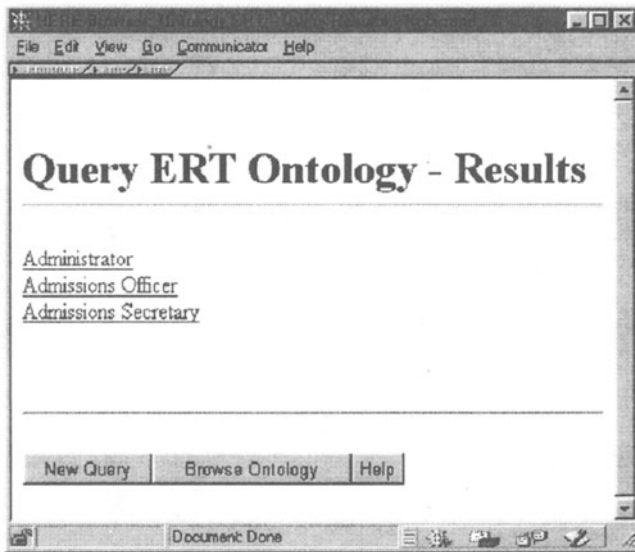


Figure 15: Displaying query results.

In the window above we can view the results of a very simple query requesting all the entities contained in an ERT model whose “Entity Name” starts from “A”. The queries can actually be much more complicated, they are always specified in OQL (Object Query Language), as defined by ODMG, and they are submitted from the clients to the WWW server via the internet.

8. CONCLUSIONS

The main idea behind the HERE project was to comprehend some of the main issues-problems in requirements engineering and provide a hypermedia environment able to address these issues. Our research established our own views regarding the requirements engineering process and issues and resulted to a

theoretical framework which is tight related with its implemented HERE environment counterpart.

HERE attempts to provide logical links amongst models, designs, requirement specifications, and other informal information and thus, it allows reasoning for the impact of one part of the system development knowledge to another and the correlation between specifications (possibly formal) and their actual (possibly informal) sources. More specifically the HERE environment addresses the following issues:

- i) Linking enterprise models, information system specifications and raw requirements. This makes possible to reason about changes required in the information system as a consequence of changes of the enterprise.
- ii) Improving the communication between system developers and clients because raw requirements are visibly linked to system requirements.
- iii) Improving the traceability and validation of requirements
- iv) Facilitating the integration of viewpoints and representations within an organisation through the establishment and
- v) Facilitating the linking of requirements to the human sources and other requirements-related information.

The HERE framework aims at the Problem Domain Information Integration with the use of Ontologies:

- Ontologies provide a neutral medium and unified representation towards the integration of the various aspects of problem domain knowledge as described above.
- Ontologies provide a mechanism for expressing the interrelationships between knowledge elements of the problem domain forming the basis for defragmentation of information, coherence, and traceability.
- Ontologies provide a more solid basis for representing the domain knowledge as they emphasise on the agreement dimension.

The HERE environment can be thought of as web based client-server information system for supporting the early phases of system development.

The functionality of the HERE environment can be summarised into the following:

- The ability to specify, manipulate, and manage Ontology Specifications of Domain Models and cross reference links among them using the ontology formalism.

- The ability to browse these models and relative knowledge in a hypermedia setting supported by the underlined object-oriented multimedia database. This means that the HERE user can access both formal models and their original informal sources in their native form.
- The ability to browse, retrieve and manipulate instances of these models and links among them.

HERE approach is generic enough to accommodate users from various backgrounds, like requirements engineers, system analysts, designers, implementers, end users and generally all stakeholders involved in the process.

Our current work status includes the development of Ontologies such as the ERT ([Theodoulidis et al, 1991]), Enterprise ([Fraser, 1994]), and the Non-Functional Requirement Framework ([Chung et al, 1994]) Ontologies, whilst future work will target at the further refinement of the Ontology development process and specification. In implementation terms, the refinement of our prototype will increase its functionality, allowing for more interaction, e.g. graphical capabilities, etc.

Finally, it is expected that the HERE framework will enhance the system development process, especially in the early stages of development, by reducing the complexity of information encapsulated in the RE products.

REFERENCES

- [ARIANE5, 1996] ARIANE 5 Flight 501 Failure Report by the Inquiry Board
The Chairman of the Board : Prof. J. L. LIONS E.S.A. (European Space Agency) Public Relations Press Releases and Information Notes, Paris, 19 July 1996.
- [Bigelow, 1988] Bigelow J., "Hypertext and CASE" IEEE Software, pp 23-27, March 1988.
- [Boehm, 1989] Boehm, B.W. "Software Risk Management" IEEE Computer Society Press: Washington, 1989.
- [Chung et al, 1994] Chung L., Nixon B. A. and Yu E. "Using Non-Functional Requirements to Systematically Support Change" ISRE 94.
- [Conklin & Begeman, 1988] Conklin J., Begeman M.L., "gIBIS: A Hypertext Tool for Exploratory Policy Discussion", ACM Transactions on Office Information Systems, Vol. 6, No. 4, pp303-331, October 1988.
- [Farquhar et al, 1995] Farquhar A., Fikes R., Pratt W., Rice, J. "Collaborative Ontology Construction for Information Integration" Stanford University WWW archive, source: <http://www-ksl-svc.stanford.edu:5915/>, 1995.

- [Fraser, 1994] Fraser, J. (1994) "Managing Change through Enterprise Models" In Milne, R., Montgomery, A. (Eds) Applications and Innovations in Expert Systems II, SGES Publications, 1994.
- [Gough et al, 1995] Gough, P. A., Fodermiski F. T., Higgins S. A., Ray S. J., "Scenarios - an Industrial Case Study and Hypermedia Enhancements" RE'95, IEEE Computer Society Press, Los Alamitos, California, York, England, 1995, pp.10-17.
- [Gruber, 1993] Gruber T. "A translation approach to portable ontology specifications" Knowledge Acquisition, 5, pp199-220, 1993.
- [Guarino & Giaretta, 1995] "Ontologies and knowledge bases: Towards a terminological clarification" KB&KS'95 Conference, University of Twente, The Netherlands 1995.
- [IWSSD-8, 1996] Proceedings of the 8th International Workshop on Software Specification and Design (IWSSD-8), Schloss Velen, Germany, 22-23 March 1996.
- [Karakostas & Loucopoulos, 1989] Karakostas V., Loucopoulos P., "Constructing and Validating Conceptual Models of Office Information Systems: a Knowledge-Based Approach" Proc. 3rd Conference Putting into practice Methods and Tools as aids to design Information Systems, Nantes, 1989.
- [LAS, 1993] Report on the London Ambulance Service (LAS) inquiry, February 1993, version 0.99, electronically prepared by Antony Finkelstein, 30-11-95, original ISBN no: 0 905133 70 6.
- [Loucopoulos & Karakostas, 1995] Loucopoulos P., Karakostas V., "System Requirements Engineering" McGraw-Hill International series in software engineering, 1995.
- [Macaulay, 1996] Macaulay, Linda. "Requirements engineering : with 80 figures / Linda A. Macaulay. - London : Springer, 1996. - (Applied computing).
- [Nuseibeh et al, 1994] Nuseibeh B., Kramer J., Finkelstein A. "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification" IEEE Transactions on Software Engineering, October 1994.
- [O2, 1995] O2C Reference Manual, Release 4.6, November 1995.
- [ODMG, 1997] Edited by R.G.G. Cattell, Douglas Barry, Dirk Bartels, Mark Berler, Jeff Eastman, Sophie Gamerman, David Jordan, Adam Springer, Henry Strickland, and Drew Wade "The Object Database Standard: ODMG 2.0" Morgan Kaufmann Publishing, ISBN 1-55860-463-4, 1997.
- [POET 1] POET Software, <http://www.poet.com/>.
- [Pohl, 1994] Pohl K., "The three dimensions of requirements engineering: A Framework and its applications" Information systems Vol. 19, No. 3, pp. 243-258, 1994.
- [Ramamoorthy et al, 1984] Ramamoorthy C.V., Prakash A., Tsai W.T., Usuda Y. "Software Engineering: Problems and Perspectives". IEEE Computer, pp. 191-209, October, 1984.

- [Schreiber et al, 1995] A.Th. Schreiber, B. J. Wielinga, and W. H. J Jansweijer. "The KACTUS view on the 'O' word." Technical Report, ESPRIT Project 8145 KACTUS, University of Amsterdam, 1995.
- [SEI, 1995] "Systems Engineering Capability Maturity Model, Version 1.1, A" Software Engineering Institute report CMU/SEI-95-MM-003, November 1995.
- [Theodoulidis et al, 1991] Theodoulidis, C., Loucopoulos P., Wangler, B. "A Conceptual Modelling Formalism For Temporal Database Applications" Information Systems, 16(4): p401-416, 1991.
- [Uschold, 1995] Uschold M. "Towards a Methodology for Building Ontologies" Workshop on Basic Ontological Issues in Knowledge Sharing, held in conduction with IJCAI-95, 1995.
- [Wielinga et al, 1994] B. Wielinga, A. Th. Schreiber, W. Jansweijer, A. Anjewierden, and F. van Harmelen. "Framework and formalism for expressing ontologies." ESPRIT Project 8145 KACTUS, deliverable DO1b.1, UvA, 1994.
- [Wood et al, 1994] Wood D. P., Christel M. G., Scott S. M. "A Multimedia Approach to Requirements Capture and Modeling" Proceedings of the first International Conference on Requirements Engineering, Colorado Springs, Colorado, April 18-22, 1994.

BIOGRAPHY

Mr. V. *Papaioannou* graduated in 1992 from the Department of Physics, University of Ioannina Greece and completed his M.Sc. Course in Computation, UMIST, in 1994. He is now a Ph.D. Candidate in the Department of Computation, UMIST, under the supervision of Dr. B. Theodoulidis. He is a member of ACM, IEEE and the Requirements Engineering Specialist Group of the BCS.

Dr. B. *Theodoulidis* holds a Diploma in Computer Engineering and Informatics from the University of Patras, Greece, an M.Sc. in Computer Science from the University of Glasgow, United Kingdom and a Ph.D. in Computation from the University of Manchester Institute of Science and Technology. He is currently a senior lecturer in the Department of Computation at UMIST where he has been working since January 1989.

He is Member of the British Computer Society, Member of the ACM, Member of IEEE Computer Society, Fellow Member of the Greek Computer Society and Fellow Member of the Engineering Council of Greece. His research interests lie in the areas of Databases, Requirements Engineering, System Development Methodologies and Computer-Aided Software Engineering. He has extensively published in these areas and his research work has been supported by the

Engineering and Physical Sciences Research Council (EPSRC), the ESPRIT programme of the European Union and by British and European industry.

Dr Babis Theodoulidis currently coordinates the activities of the TimeLab research laboratory specialising on intelligent information systems engineering.